

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-100939

(43)Date of publication of application : 23.04.1993

(51)Int.Cl.

G06F 12/00

(21)Application number : 03-213036

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 31.07.1991

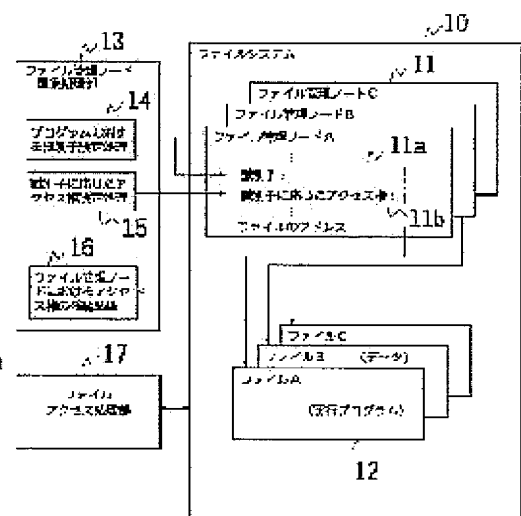
(72)Inventor : HAYATA HIROSHI

(54) FILE SYSTEM

(57)Abstract:

PURPOSE: To execute read-out and write of a file only from a specific program by deciding an identifier of a program by an identifier of a file management node, and executing the access management by the access right corresponding to the identifier.

CONSTITUTION: An access right setting means 13 sets an identifier 11a given to a program of a file 12 as file management information to a file management node 11 for managing the file 12. Also, the access right 11b corresponding to the identifier 11a is registered and set as the access right of the file 12. In such a way, in the case of accessing the file 12 by executing the program, a file access managing means 17 decides an identifier of the program concerned by the identifier 11a set to the file management node 11. Subsequently, by this identifier, the access right 11b registered in the file management node 11 of the file 12 being an access object is discriminated. In accordance with information of this access right 11b, an access of the file 12 is controlled.



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-100939

(43)公開日 平成 5 年(1993) 4 月23日

(51)Int.Cl.⁵

G 0 6 F 12/00

識別記号

庁内整理番号

F I

技術表示箇所

5 3 7 A 7832-5B

審査請求 未請求 請求項の数1(全 9 頁)

(21)出願番号 特願平3-213036

(22)出願日 平成 3 年(1991) 7 月31日

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂三丁目 3 番 5 号

(72)発明者 早田 宏

神奈川県川崎市高津区坂戸100番 1 号 K S

P/R&D ビジネスパークビル 富士ゼロ
ックス株式会社内

(74)代理人 弁理士 南野 貞男 (外 2 名)

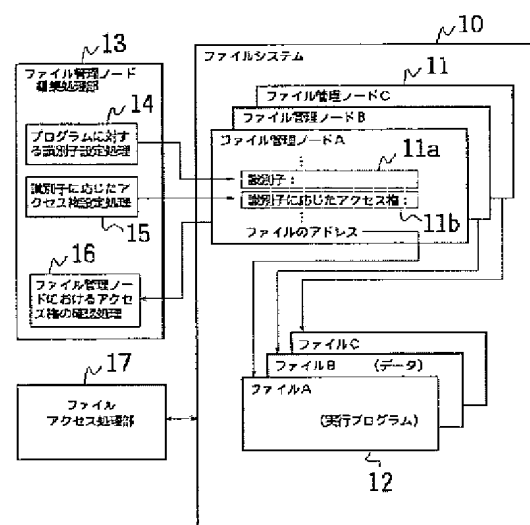
(54)【発明の名称】 ファイルシステム

(57)【要約】

【目的】 ある特定のプログラムからのみ、ファイルの読出し、ファイルへの書込みを可能とするファイルシステムを提供する。

【構成】 ファイル対応のファイル管理ノードに当該ファイルのアクセス権を登録し、ファイル管理ノードに登録したアクセス権により、ファイルアクセスを行うファイルシステムにおいて、ファイル管理ノードに当該ファイルのプログラムに与える識別子と、識別子対応のアクセス権とを登録し、プログラム実行によりファイルをアクセスする場合、ファイル管理ノードの識別子により当該プログラムの識別子を判定し、当該プログラムの識別子により、アクセス対象のファイルのファイル管理ノードの識別子に対応して設定されたアクセス権により、当該ファイルのアクセス管理を行う。

図 1



【特許請求の範囲】

【請求項1】 各々のファイル対応に設けられるファイル管理ノードに当該ファイルのアクセス権を登録し、ファイル管理ノードに登録したアクセス権により、各々のファイルのアクセスを行うファイルシステムにおいて、ファイル管理ノードに、当該ファイルのプログラムに与える識別子と当該ファイルのアクセス権として更に識別子対応のアクセス権とを登録するアクセス権設定手段と、プログラムの実行によりファイルをアクセスする場合に、ファイル管理ノードの識別子により当該プログラムの識別子を判定し、当該プログラムの識別子により、アクセス対象のファイルのファイル管理ノードに登録された識別子に対応して設定されたアクセス権により、ファイルのアクセスを管理するファイルアクセス管理手段とを含むことを特徴とするファイルシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ファイルシステムに関し、特に、情報処理装置におけるファイルシステムにおいて、アクセス権によるファイル管理機能を有効利用してシステムのセキュリティを高めたファイルシステムに関するものである。

【0002】

【従来の技術】 従来、情報処理システムにおいて、ある目的を持ったデータの集まりはファイルとして取り扱われ、データ処理がなされる。ファイルはシステム規模が大きくなると、爆発的に増加する。このため、多くの各種のファイルを統一的に取り扱うための手法が開発されている。例えば、ファイル管理は、情報処理装置で取り扱われる各種のファイルを標準的な方法で統一的に管理し、プログラムが簡便な使い方でファイルに関する処理を効率よく、経済的に行える機能を提供する。このようなファイル管理の機能は、オペレーティングシステムの中におけるファイルシステムとして提供される。プログラムは、オペレーティングシステムが提供するファイルシステムのインタフェースを介して、ファイルへの読み出しや書き込みを行うことになる。その場合、各々のファイルは、アクセス権によるファイル管理が行なわれ、データ保護、システムの機密保護などが機能できるようになっている。

【0003】 例えば、UNIXシステムにおけるファイルシステムでは、ファイルからのデータの読み出しは、readシステムコールで行なわれ、また、ファイルへのデータの書き込みは、writeシステムコールで行なわれる (Maurie J Bach著/坂本文・多田好克・村井純 訳 “UNIXカーネルの設計”，1991年6月10日，共立出版発行，pp51～54，pp82～87などを参照)。

【0004】 このようなファイルシステムにおいては、ユーザのファイルアクセスリクエストに対してのファイ

ルへの読み出しや書き込みの制御は、ファイルに対するアクセス権で管理されている。ファイルのアクセス権に関する情報はiノード(ファイル管理ノード)に設けられ、このiノードにおけるファイル管理情報により管理される。図6はファイル管理ノードであるiノードの一例を説明する図である。iノードは次のようなフィールドから構成される。

ファイル所有者識別子：所有者は個人所有者と「グループ」所有者が分け持ち、ファイルにアクセスする権利を持つ所有者を定義する。

ファイルの種類：ファイルは通常型、ディレクトリ、文字型またはブロック特殊ファイル、FIFO(パイプ)のいずれかである。

ファイルへのアクセス許可：システムは、ファイルの所有者、ファイルのグループ所有者、その他の利用者の3つの等級に従ってファイル保護を行う。各等級に対して当該ファイルの読出し(r)，書き込み(w)，実行(x)に関するアクセス権を持ち、個々に設定する。例えば、ディレクトリのファイルは、実行できないため、ディレクトリに対する実行許可では、当該ディレクトリの中でファイル名を探す権利を有することを意味する。

ファイルへのアクセス時刻：ファイルを最後に更新した時刻，最後にアクセスした時刻，iノードを最後にアクセスした時刻を示す。

ファイル内のデータにディスクアドレスに関するアドレス表：利用者はファイル中のデータをバイトの論理ストリームとして扱うが、システムのカーネルはデータを不連続なディスクブロックとして管理する。iノードはファイルのデータを含むディスクブロックを識別する。

ファイルの大きさ：ファイル中のデータは、バイト0から始まるファイルの最初から数えたバイト数でアドレス指定することができる。このファイルの大きさは、ファイル中のデータの最高のバイト変位よりも1だけ大きい。例えば、利用者があるファイルを作成し、ファイルのバイト変位1000のところに1バイトのデータを書込んだ場合、ファイルの大きさは1001バイトとなる。

【0005】 例えば、図6に示すiノードの例は、“MJB”が所有する通常型のファイルのiノードの例である。このファイルは6030バイトのデータを含んでおり、許可モード(アクセス権)として“rwxr-xr-x”の9桁の文字データを設定している。ここでの最初の3桁の文字“rwx”により、ファイルシステムは所有者“MJB”に対して、ファイルの読出し、書き込み、実行を許可していることを意味している。また、次の3桁の文字“r-x”により、“OS”というグループのメンバーに対し、ファイルシステムは当該ファイルの読出しと実行のみを許可していることを意味し、そして、最後の3桁の文字“r-x”により、他の利用者に対して、ファイルシステムは当該ファイルの読出しと実

行のみを許可することを意味している。。このため、
“OS”というグループのメンバーと他の利用者は、当該ファイルに対して、ファイルの読出しと実行だけが可能であり、書込みはできない。

【0006】また、iノードでは、最終アクセス時刻、最終更新時刻などの時刻情報を保持して、ファイルを管理している。この例のiノードでは、最後に誰かがこのファイルを読み出したのは1990年10月23日午後1時45分であり、最後に誰かがこのファイルに書込みをしたのは1990年10月22日午後10時30分であるという管理情報が保持されている。

【0007】このように、UNIXシステムのファイルシステムでは、各々のファイルに1対1に設けられたファイル管理ノード(iノード)を用い、そのファイル管理ノードに当該ファイルのアクセス権、所有者などのファイル管理情報を設定し、当該ファイルを管理している。

【0008】

【発明が解決しようとする課題】ところで、ファイルシステムでは、上述のように、ファイル管理ノードに設定する当該ファイルのアクセス権、所有者などのファイル管理情報により、当該ファイルが管理されているため、利用者がアクセス権さえ、何らの方法により持てば、同じファイルを複数のプログラムから読み出したり、書込んだりできることになる。このようなファイルシステムを用いて、例えば、データベース管理システムのような特定のプログラムからのみファイルへの読み出しや書込みを行い、一般のプログラムからは読み出しのみしか行えないようなシステムを構成する場合には、上述のようなファイル管理機能では、その対応のプログラムを実現する上で不具合が生ずることになる。

【0009】本発明は、上記のような問題点を解決するためになされたものであり、本発明の目的は、ある特定のプログラムからのみファイルの読出し、ファイルへの書込みを可能とするファイルシステムを提供することにある。

【0010】

【課題を解決するための手段】上記の目的を達成するため、本発明のファイルシステムは、各々のファイル対応に設けられるファイル管理ノード(11;図1)に当該ファイル(12;図1)のアクセス権を登録し、ファイル管理ノードに登録したアクセス権により、各々のファイルのアクセスを行うファイルシステムにおいて、ファイル管理ノード(11;図1)に、当該ファイルのプログラムに与える識別子と当該ファイルのアクセス権として更に識別子対応のアクセス権とを登録するアクセス権設定手段(13;図1)と、プログラムの実行によりファイルをアクセスする場合に、ファイル管理ノードの識別子により当該プログラムの識別子を判定し、当該プログラムの識別子により、アクセス対象のファイルのフ

イル管理ノードに登録された識別子に対応して設定されたアクセス権により、ファイルのアクセスを管理するファイルアクセス管理手段(17;図1)とを含むことを特徴とする。

【0011】

【作用】ファイルシステムにおいては、各々のファイル対応に設けられるファイル管理ノード(11)に当該ファイル(12)のアクセス権を登録し、ファイル管理ノードに登録したアクセス権によって、各々のファイルのアクセス権が管理され、ファイルのアクセス制御が行なわれる。このようなファイルシステムにおいて、アクセス権設定手段(13)と、ファイルアクセス管理手段(17)とが設けられる。アクセス権設定手段(13)は、ファイルを管理するためのファイル管理ノード(11)に、ファイル管理情報として、当該ファイルのプログラムに与える識別子を設定し、更にファイルのアクセス権として、識別子対応のアクセス権とを登録設定する。これにより、ファイルアクセス管理手段(17)は、プログラムの実行によりファイルをアクセスする場合、ファイル管理ノードに設定した識別子により当該プログラムの識別子を判定し、当該プログラムの識別子により、アクセス対象のファイルのファイル管理ノードに登録された識別子に対応して設定されたアクセス権を判別し、当該アクセス権の情報に従って、ファイルのアクセスを行うアクセス制御を行う。

【0012】このように、実行プログラムのファイルからは、プログラム実行にかかるファイルアクセス要求が発行された場合、当該プログラムの識別子が判定され、その識別子に対応して設定されているアクセス権によりファイルアクセス制御が行なわれる。これにより、単にファイル所有者、利用者に対して設定されているアクセス権によるファイルアクセス制御のみでなく、実行プログラムのレベルでのアクセス権でのファイルのアクセス制御が可能となり、ファイル操作、ファイル処理、ファイル管理などシステム構築の自由度が大きくなり、また、システムの安全性を配慮したシステム構成が容易に実現可能となる。

【0013】

【実施例】以下、本発明の一実施例を図面により具体的に説明する。図1は本発明の一実施例にかかるファイルシステムの要部構成を説明するブロック図である。図1において、10はファイルシステム、11は各々のファイル管理ノード、12は各々のファイルを示している。各々のファイル12とファイル管理ノード11とは1対1に対応している。ファイルAに対してはファイル管理ノードAが対応し、ファイルBに対してはファイル管理ノードBが対応し、また、ファイルCに対してはファイル管理ノードCが対応している。ファイル管理ノード12には自己が管理する該当のファイルにおける実行プログラムに対して、識別子を設定するため識別子フィール

ド11aと、識別子に応じたアクセス権を設定するための識別子アクセス権フィールド11bが設けられている。

【0014】このようなファイル管理ノード12に対して、識別子、識別子に応じたアクセス権などを個別に設定し、また、設定したファイル管理情報の確認を行うため、ファイル管理ノード編集処理部13が設けられる。このファイル管理ノード編集処理部13の処理機能により、プログラムに対する識別子設定処理14、識別子に応じたアクセス権設定処理15、ファイル管理ノードにおけるアクセス権確認処理16などが行なわれる。

【0015】また、このように設定されたファイル管理ノードにおけるファイル管理情報を用いて、ファイルアクセス処理を行う場合のファイルアクセス制御を行うため、ファイルアクセス処理部17がシステム内に設けられる。

【0016】図2は、ファイルシステムにおけるファイル管理ノードと各ファイルの関係をファイル管理情報のデータ例と共に説明する図である。データファイルのファイル管理ノードの例を図2(A)に示し、実行プログラムファイルのファイル管理ノードの例を図2(B)に示している。各ファイル管理ノードは、従来のファイルシステムにおけるファイル管理ノードと同様に、ファイル所有者、ファイル所有者のグループ、ファイルの最終アクセス時刻、ユーザに応じたアクセス権、ファイルの実体のディスク上の位置を示すディスクのアドレスなどのファイル管理情報を保持しており、ここでは、更に、プログラムに与えられる識別子、プログラムに応じたアクセス権のファイル管理情報が付加される。

【0017】ファイル内容がデータであるファイル21に対するファイル管理ノード20には、ファイル管理情報として、所有者“Hayata”，グループ“FXKSP”，最終アクセス時刻“Apr. 5 1991 19:00:00”，最終変更時刻“Apr. 4 1991 12:30:00”，ユーザに応じたアクセス権“rwxr-xr-x”，プログラムに応じたアクセス権“(100rwx)(101r-)(102r-x)”，プログラムに与えられる識別子“0”，ディスクのアドレス“12345”が設定されている。

【0018】ファイル内容が実行プログラムであるファイル23に対するファイル管理ノード22には、ファイル管理情報として、所有者“Hayata”，グループ“FXKSP”，最終アクセス時刻“Apr. 3 1991 19:00:00”，最終変更時刻“Apr. 3 1991 12:30:00”，ユーザに応じたアクセス権“rwxr-xr-x”，プログラムに応じたアクセス権“0”，プログラムに与えられる識別子“100”，ディスクのアドレス“22345”が設定されている。

【0019】この例では、データファイルのファイルA

(21)に関して、そのファイル管理情報であるプログラムに応じたアクセス権として、“(100rwx)(101r-)(102r-x)”が設定されている。この設定のプログラムに応じたアクセス権の意味は、識別子100のプログラムについては、読出し、書込み、実行を許可し、識別子101のプログラムについては、読出しのみを許可し、また、識別子102のプログラムについては、読出し、実行を許可し、書込みは許可しない。それら以外のプログラムについては、読出しも、書込みも、実行も許可しないことを意味している。なお、ファイルAの識別子フィールドは“0”となっており、実行形式ファイルの実行プログラムファイルでないため、ファイルAには識別子は与えられていない。

【0020】また、実行プログラムファイルのファイルBに関しては、そのファイル管理情報であるプログラムに与えられる識別子として“100”が設定されており、このファイルBにおけるプログラムには識別子100が与えられることを示している。また、ファイルBは、データファイルではないので、プログラムに応じたアクセス権のファイル管理情報は設定されておらず、当該フィールドの各々の識別子に応じたアクセス権の情報は与えられていない。

【0021】図3は、ファイル管理ノードのファイル管理情報を用いてファイルアクセス時に行なわれるアクセス権チェック処理の一例を示すフローチャートである。この処理は、ファイルアクセス処理部(17;図1)により行なわれる。このアクセス権チェック処理では、まず、ステップ31において、実行プログラムファイルに対するファイル管理ノードを得ると、次に、ステップ32において、ファイル管理ノードからプログラムに与えられた識別子IDを得る。次に、ステップ33において、読み出し対象ファイルのファイル管理ノードを得る。そして、次のステップ34において、ファイル管理ノードからプログラムに応じたアクセス権データAを読み出す。次に、ステップ35において、読み出したアクセス権データAの中からプログラム識別子IDに対応するアクセス権ACを得る。そして、次のステップ36において、アクセス権ACの内容の判別を行い、アクセス権に応じたアクセス処理を行う。すなわち、アクセス権ACにread許可がある場合には、当該ファイル読出しが可能なので、リターン処理を行い、ファイルアクセスを行っているREADシステムコールのメインルーチンに戻る。アクセス権ACにread許可がない場合には、当該ファイル読出しが不可なので、エラーリターン処理を行い、ファイルのリードエラー処理を行う。

【0022】このようにして、プログラムの実行中にファイルがアクセスがなされた場合、当該実行プログラムに与えられている識別子に対応のファイル管理ノードから得て、この識別子よりアクセス対象のファイル管理ノードから、識別子対応のアクセス権(プログラムに応じ

たアクセス権)を得て、このアクセス権により、ファイルアクセスを行うファイル管理を行う。これにより、アクセス権情報によるアクセス管理は、単にファイル所有者、利用者に対して設定されているアクセス権によるファイルアクセス制御のみでなく、実行プログラムのレベルでのアクセス権でのファイルのアクセス制御が可能となる。また、ファイル処理、ファイル操作にかかるシステム構築の自由度が大きくなり、システムの安全性を配慮したシステム構成が容易に実現可能となる。

【0023】次に、このようなファイルシステムに用いられるファイル管理ノードにおけるファイル管理情報を設定し、確認するための処理機能要素について説明する。前述したように、ここでは、ファイル管理ノードに対して、識別子、識別子に応じたアクセス権などを個別に設定し、また、設定したファイル管理情報の確認を行うため、ファイル管理ノード編集処理部(13;図1)が設けられている。このファイル管理ノード編集処理部の各々の処理機能により、プログラムに対する識別子設定処理、識別子に応じたアクセス権設定処理、ファイル管理ノードにおけるアクセス権確認処理などが行なわれる。

【0024】図4はファイル管理ノードに対するプログラム識別子設定処理を示すフローチャートであり、また、図5はファイル管理ノードに対するプログラム対応のアクセス権設定処理を示すフローチャートである。例えば、図4に示すファイル管理ノードに対するプログラム識別子設定処理では、まず、ステップ41において、ファイル名から対応するファイル管理ノードを得て、次のステップ42で、このファイル管理ノードに対してプログラムに与える識別子をセットする。具体的には、例えば、ファイル毎のファイル管理ノードに、当該ファイルの識別子を設定する手続き関数として、次のような関数形式のプログラムset_idを作成して実行する。

set_id(ファイル名, 識別子)

set_idは、実行プログラムであるファイル名ならびに識別子を引数としてとり、指定したファイル名のファイル管理ノードに指定した識別子を書き込む処理を行う手続き関数である。

【0025】また、図5に示すファイル管理ノードに対するプログラム対応のアクセス権を設定する処理では、まず、ステップ51において、ファイル名から対応するファイル管理ノードを得て、次のステップ52において、このファイル管理ノードに対して、識別子とそれに応じたアクセス権データをセットする。具体的には、例えば、ファイル毎のファイル管理ノードに対し、識別子(プログラム)に応じたアクセス権を設定する手続き関数として、次のような関数形式のプログラムchapmodを作成して実行する。

chapmod(ファイル名, 識別子, アクセス権)

chapmodは、ファイル名, 識別子ならびにアクセス権を

引数として取り、指定したファイル名に対応するファイル管理ノードに、指定した識別子に応じとそれに対応したアクセス権の情報を書き込む処理を行う手続き関数である。

【0026】また、ファイルアクセスを行う上でのファイル毎の各々の識別子に応じたアクセス権を確認する機能コマンドは、ファイルの読出し、書き込みなどのファイルアクセスを行うreadや、writeなどのシステムインタフェース機能を用いることにより実行する。すなわち、システムにおけるファイルインタフェース機能を用いて、従来からユーザ対応に設定したアクセス権の確認処理と同様にして、プログラム(識別子)に対応して設定したアクセス権の確認を行う。

【0027】以上説明したように、本実施例のファイルシステムによれば、実行プログラムのファイルに識別子を与えて、当該ファイルのプログラムに対応する識別子を設定しておき、また、アクセス対象のデータのファイルには、識別子に応じたアクセス権を与えておく。これにより、プログラム実行により、データファイルへのアクセスが行なわれる場合、実行プログラムのファイルに設定された識別子により、プログラムに設定された識別子を判定し、この識別子に基づいて、データファイルの識別子対応のアクセス権を判定する。そして、このアクセス権によりファイルアクセス制御を行う。これにより、ファイル管理を、ユーザレベルだけでなく、プログラムレベルにおいても同様に行うことができる。また、プログラム毎に一意の識別子を与えることにより、特定のプログラムからのみのアクセスの制御を可能とするファイルが実現できる。

【0028】

【発明の効果】以上に説明したように、本発明によれば、実行プログラムのファイルからは、プログラム実行にかかるファイルアクセス要求が発行された場合、ファイル管理ノードから当該プログラムの識別子が判定され、データファイルのファイル管理ノードにその識別子に対応して設定されているアクセス権によりファイルアクセス制御が行なわれる。これにより、単にファイル所有者、利用者に対して設定されているアクセス権によるファイルアクセス制御のみでなく、実行プログラムのレベルでのアクセス権でのファイルのアクセス制御が可能となる。また、ファイル操作、ファイルの管理などのシステム構築の自由度が大きくなり、システムの安全性を配慮したシステム構成が容易に実現可能となる。

【図面の簡単な説明】

【図1】 図1は本発明の一実施例にかかるファイルシステムの要部構成を説明するブロック図、

【図2】 図2はファイルシステムにおけるファイル管理ノードと各ファイルの関係をファイル管理情報のデータ例と共に説明する図、

【図3】 図3はファイル管理ノードのファイル管理情

報を用いてファイルアクセス時に行なわれるアクセス権チェック処理の一例を示すフローチャート、

【図4】 図4はファイル管理ノードに対するプログラム識別子設定処理を示すフローチャート、

【図5】 図5はファイル管理ノードに対するプログラム対応のアクセス権

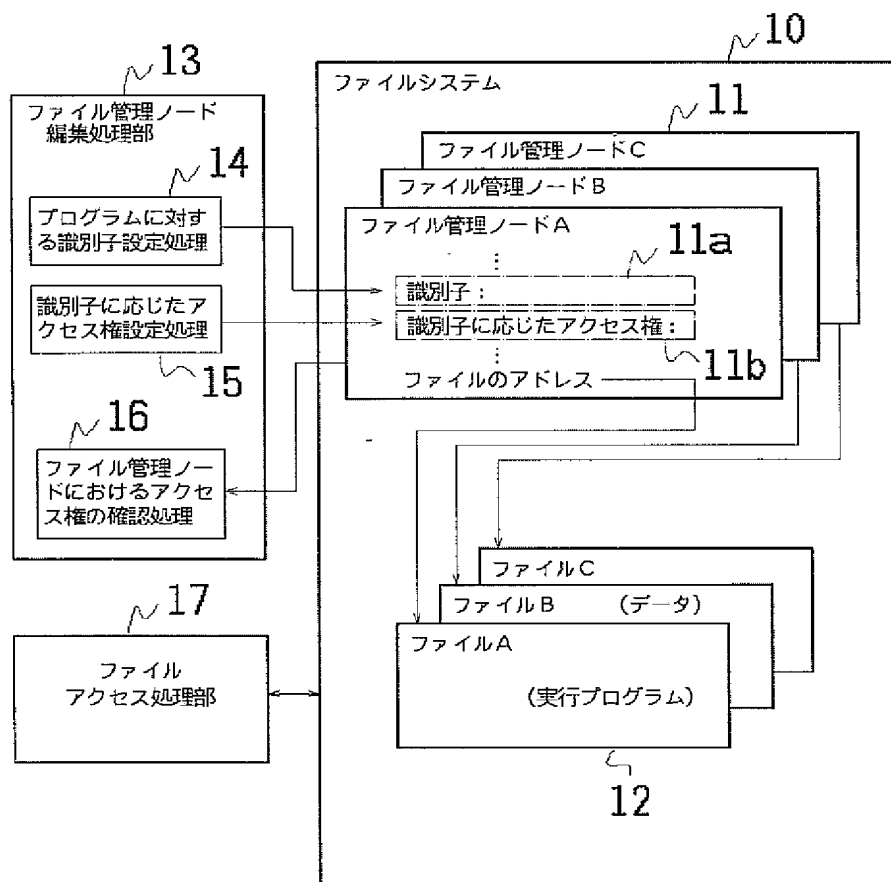
【図6】 図6はファイル管理ノードであるiノードの一例を説明する図である。

* 【符号の説明】

10…ファイルシステム、11…ファイル管理ノード、11a…識別子フィールド、11b…識別子アクセス権フィールド、12…ファイル、13…ファイル管理ノード編集処理部、17…ファイルアクセス処理部、20…ファイル管理ノードA、21…ファイルA（データファイル）、22…ファイル管理ノードB、21…ファイルB（実行プログラムファイル）。

【図1】

図1



【図2】

図2 (A)

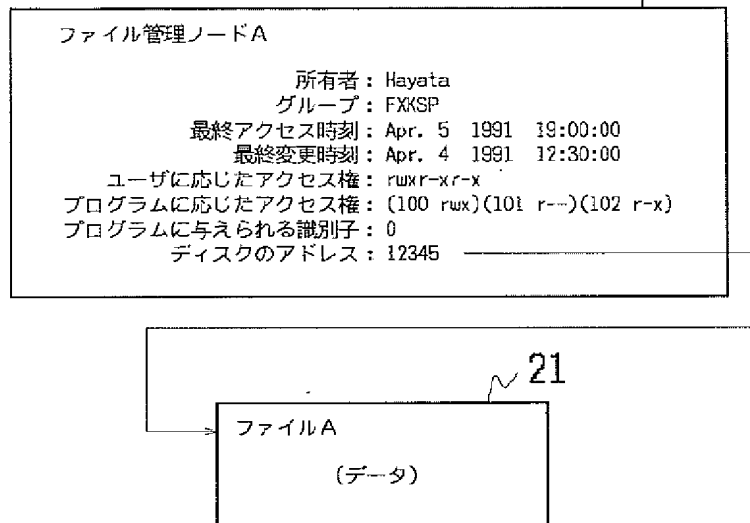
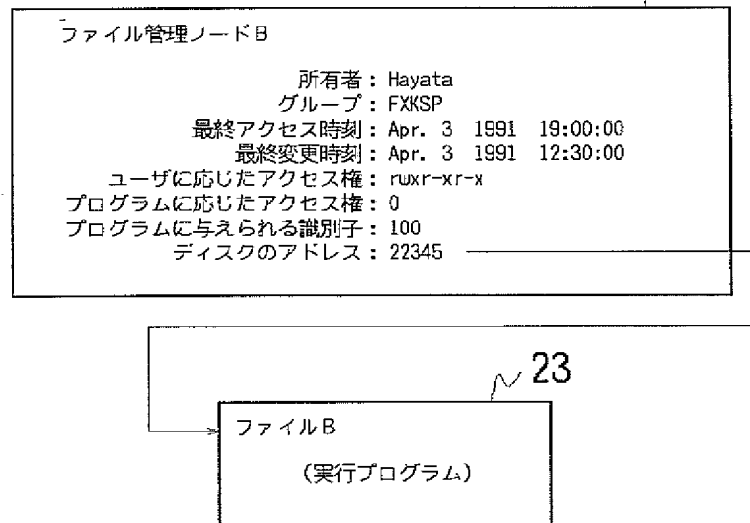
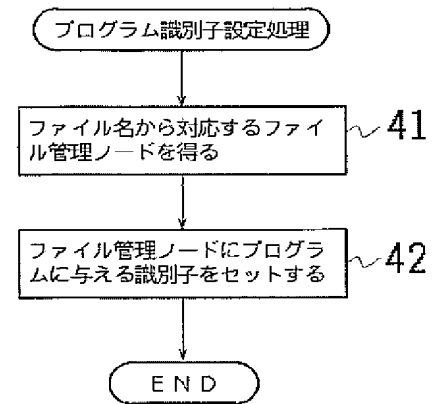


図2 (B)



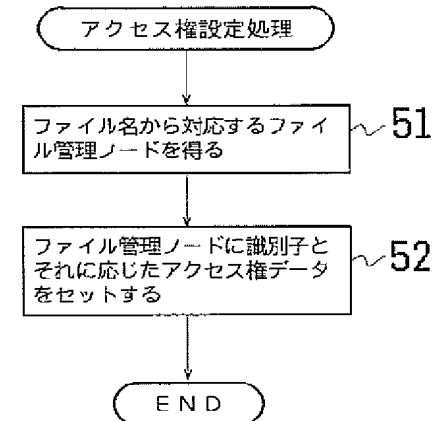
【図4】

図4



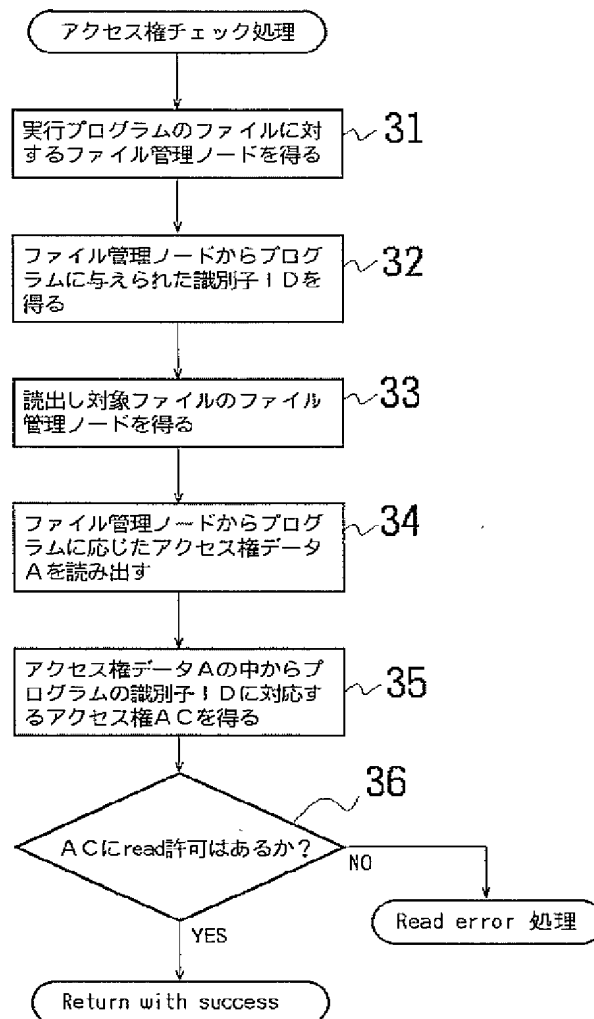
【図5】

図5



【図3】

図3



【図6】

図6

i ノード	
所有者 :	MJB
グループ :	OS
ファイル種類 :	通常ファイル型
許可モード :	rwxr-xr-x
最終アクセス時刻 :	Oct. 23 1990 1:45 P.M.
最終変更時刻 :	Oct. 22 1990 10:30 A.M.
i ノードの最終更新時刻 :	Oct. 23 1990 1:30 P.M.
大きさ :	5030バイト
ディスクのアドレス :	

